

Backend Engineering Challenge

Challenge Prompt

Create a system that emulates a RESTful API (including data model and the backing implementation) for money transfers between accounts.

You must use Go programming language to demonstrate your best work. The datastore also be part of this Go project and must run in-memory for the sake of this test – though it does not need to be persistent over restarts.

Accounts

You can download a JSON file of accounts structures from <https://git.io/Jm76h>. Accounts must be parsed from the file and ingested into your system. when all accounts have been consumed, your system should state it is ready to make a transfer through a console output message. You must provide an endpoint to list all accounts being ingested, the balance field should always reflect the result of a transfer.

Transfer requests should be validated to confirm both from and to account, and that the transfer will not result in a negative balance. Appropriate response message to be given as needed.

Grading Rubric

You are expected to build a system to handle the above scenarios that fulfills each element of this grading rubric to the best of your ability.

- Meets all specified requirements from Challenge Prompt above.
- Is valid runnable code via CLI (should not require a pre-installed container/server)
- Has appropriate usage of concurrency, and data structures.
- Has extendable architecture.
- Demonstrate with tests that the API works as expected.
- Has production-quality code cleanliness.
- Has production-quality docs on all public functions.
- Has a README file that contains:
 - Instructions on how to run and test your code in a local environment.
 - Any other design choices you would like the interviewers to know.

Code Submission

Create a private repo on Github and invite : subs@paytabs.com as a collaborator.